Digitization Errors In Hungarian Documents

Máté Pataki[1] Tamás Füzessy[2]
[1]*Department of Distributed Systems*
Computer and Automation Research Institute of the Hungarian Academy of Sciences
[2]*FreeSoft Nyrt.*
mate.pataki@sztaki.hu, tfuzessy@freesoft.hu

Abstract. Our task was to analyze a certain digitizing system, check what type of errors emerge during the process, and how these errors effect the searchability of the digitized documents. We have set up a testbed which is suitable for the automatic processing of digitized texts in a large scale. In this paper we shortly introduce the methodology of document digitization emphasizing the error-sources in the process, and sketch the results obtained from our test-system, especially the Hungarian language dependent characteristics of the emerging errors.

Keywords: Character recognition, text processing, search, error, OCR

1 Introduction

Digitizing printed texts is a required process. The reason behind is that besides preservation purposes, the digitized image/text can be retrieved and accessed by the wide public more easily this way. For the latter not only digitizing, but also recognizing is required, so the digitized image must be translated to textual information again. Unfortunately, this process in itself contains quite a lot of possibilities of failure, and even at the state of the art it can not be accomplished without errors (where error means that the digitized text compared to the original one might be defected in its structure and/or in its content). If our aim is the retrieval of information, the structural information has lower priority than the pure content. Certainly, the structure, e.g. the structure of the paragraphs in a text, the placement of figures, etc., itself can contain important information (for example, when digitizing maps), but asking for this during a normal text-retrieval search is really not easy. In this paper we focus our attention to the usual content search functions, like "computer, retrieve the document containing the following text: ...".

2 Process of Scanning Documents

The human vision and structure recognition is a rather complex process [1]. By using artificial tools, this process can not be modeled at the complexity of the human recognition, while this is what we need for the full process of printed texts. (However, there are similarities between the character recognition of the human vision and the artificial character recognition processes.) There are fields in the human recognition processes still not mapped by researchers, and it is only one among the questions. Behind the differences in complexity, the main difference between the artificial and the human approach is the architecture, namely the basic structure of the information processing/storing units.

Fortunately, in normal cases there is no need for the complete understanding, complex processing of the information, therefore, simplified methods, which still have adequate results, can be used. The basic steps of the process [2] are detailed in the following subsections.

2.1 Sampling

Sampling is the process of converting a signal (for example, a function of continuous time or space) into a numeric sequence (a function of discrete time or space). After this step the printed information itself is in digital form, but is hardly searchable. The digitizing equipment converts the sensed image to luminance and color parameters. The main question is the density of the sampling. If we wish to completely reconstruct the original image, the Nyquist-Shannon [3] sampling theorem states that: "Exact reconstruction of a continuous-time baseband signal from its samples is possible if the signal is bandlimited and the sampling frequency is greater than twice the signal bandwidth." Yet this is not our task. The needed density in our case varies depending on the digitized source. It is completely different when we are to preserve codices for the posterity than when we are to digitize simple printed books from the 20th century. For the later one 300dpi, for smaller printed characters maybe 600dpi must be convenient. The main problem when we take too many samples is that we have to handle much more data than necessary, while when we have less samples than necessary, we will have undersampling error and e.g. the so called Moiré [4] pattern can emerge (from the spectral components of the sampled signal some will overlap causing artificial noise in the digitized image).

2.2 Quantization

During quantization the sampled signal will be converted in a way that the spectrum will be limited to certain values, e.g. in grayscale processing the result will be limited only for the luminance values of the sampled image. Some character recognition methods need only binary values, when under a threshold luminance value the image is considered to be black, while above it is white. This threshold can be chosen in an adaptive way, meaning that, based on the luminance values in the surrounding of the analyzed pixel, the threshold can

be changed dynamically. As a result of the quantization, quite a lot of important information is lost. Such can happen at the previously mentioned binary quantization with the elimination of most of the color values. When there were black colored characters with blue stamp over them, after binary quantization, the stamp can not be separated from the written text.

2.3 Preprocessing

During preprocessing the previously yielded image will be modified to suit the result for the applied optical character recognition algorithm. First, different kinds of noise removal algorithms are used to eliminate the noise of the digitizing equipment and the noise on the original content (different kinds of dust and dirt patches). Hungarian language has quite a lot of characters with accents, these accents hardly differ from some types of noise, and it can easily happen that a badly chosen or parameterized noise removal algorithm will eliminate these accents as well. Other important tasks of preprocessing are to correct the geometric distortion, separate the background from the foreground, segment and identify the layout. Usually different morphologic operators are applied (erosion, skeletonization) to separate the characters while their most important features remain the same. Contour detection, polygon-matching, etc., can be used when the different separated parts of the image are attached with feature-vectors.

2.4 Character recognition

The next step is character recognition. Though there are language independent, training-based, generic algorithms, but generally the language dependent, more efficient methods are used. The two main approaches are:

- Template-matching
  A pattern is compared to the separated sample of the analyzed character and the differences are measured
- Feature based [5]
  The feature-vectors earned during the preprocessing are compared to the feature vectors of known characters

The state of the art OCR (Optical Character Recognition) softwares use a kind of combined, hierarchic, complex approach. The result of the character recognition can be a series of characters, or in better cases, it results in probability vectors denoting the similarities of the identified characters to known characters in previously stored character sets. The main source of error in this step originates from the differences of the digitized and the stored sample character sets.

2.5 Text recognition and text processing

During text recognition and text processing the grammatical rules are matched with the results of the OCR process, and the offending, maybe erroneously

identified characters, are corrected [6, 7]. When the previous step resulted in probability vectors, these values can be used to support this one. Unfortunately, at this point we can introduce some errors into the digitization process, too. First, the grammatical rules are continuously changing. A text originating from the 18th century is constructed based on different grammar structure than a documentum from the 20th century. Another problem is, that grammar descriptions and dictionaries (e.g. for the Hungarian language) are usually not complete, and it can happen that otherwise meaningful constructions are not included in them. In ambiguous situations the system can change meaningful words to other, also meaningful, words.

3 Testbed

Our testbed consists of a database containing Hungarian documents in various forms (.rtf, .txt, .pdf and .doc), the digitizing software, which is capable of character recognition from digital image formats, and a branch of self-developed utilities. The documents were converted to images and different kinds of noise were generated over them (coffee-patches, traces of plying, noise), then the resulting images were sent to the digitizing application. The application tried to recognize the texts which resulted in digital, textual documents which were suitable to be compared to the original digital ones.

The comparison was done in two steps. First, a manual comparison took place for a small number of documents to identify error-categories, error-types. Then an automatic comparison took place for the whole database. After the later step, based on the results of the manual comparison, we evaluated our automatic methods and generated different statistics to tune the categorization of the error-types.

Based on the results of the comparison several search methods were tried so as to show their effectiveness over digitized Hungarian content.

3.1 Printing

The first step in the testsystem was to print the documents into images. So as to avoid further errors resulting from the transformation, we used loss free compressed TIFF images. Printing was done by a printer program which could print any document using the originally associated program, for example, DOC files were printed with Microsoft Word, PDF files with Adobe Acrobat Reader and so on. After all documents were printed some noise was added to them to emulate real documents used in real enviroments like governmental contracts. Figure 1 shows some of the typical noise patterns used for testing.

3.2 Quantization

After the artificial noise was added to the printed documents, a binary quantization was performed, to emulate black and white scanning, which, in most

Figure 1: Typical noise patterns generated over the documents

cases, is used for this kind of application. Figure 2 shows the largest noise pattern used in the testbed. It is a sound example for the previously mentioned quantization error. Some characters are not readable, while they were clearly visible and could be read behind the coffee-patch before the quantization.

3.3 Using the OCR Software

For text and character recognition we used the eImage OCR v5.1b application. It has a command line interface and is capable of batch processing, converting multiple input documents into multiple output documents. For testing purposes a plain text output was used, so no formatting information remained in the document.

The language of the OCR engine was set to Hungarian as only Hungarian texts were used. This is important because the engine could use this information in the text processing phase and as can be seen in the output, this also generated some errors.

3.4 Text Comparison

To be able to compare the input and the output documents the first ones had to be also converted to plain text format. The comparison was done by a self developed PHP program, which counted the differences between the documents and added them to a database. The database table consisted of four rows; the

Figure 2: A document page with the artificial noise over it

document ID, a word found in the original file, the converted version of the word in the re-digitized file, and the number of occurrences.

4 Results

As a result of the comparison our database contains which words were altered to which other word and which characters to which other characters or character series. Though the accuracy rate was quite high (around 95%, which is the expected value also mentioned in the literature), still we were able to find typical character/word changes (Table 1). In the followings we will show some typical errors/error-types.

Errors with accented characters

The first and largest group of errors related to accented Hungarian characters. As an explanation we would like to refer to the noise removal process detailed in Section 2.1. The "o"-related error-counts are in Table 2.

Punctuation mark errors

The most common errors with punctuation marks were the missing dots at the end of the sentence, and the exclamation mark which was often recognized as a letter "i".

Substitution of one character with a similar one

Table 1: The most frequent character changes

Orig	OCR	Count
M	m	124103
-	—	82358
É	e	75882
Á	a	71436
-	NULL	55990
		43263
V	v	42109
G	9	40713
,	,	40180
NULL	-	30378
O	õ	21321
Ó	o	18301
NULL		15324
Í	i	13992
”	"	13975
W	w	11401
–	-	10428

Table 2: Various “o”-related errors

Orig	OCR	Count
o	õ	21321
ó	o	18301
õ	ó	7438
Ö	ö	5831
õ	o	5689
Õ	õ	5488
o	ó	3112
ó	Ó	1361
o	ö	1213

The most common character substitutions are really interesting for future work with digitized documents (Table 3). For example when searching for words containing the letter “g”, one could also search for the same word, but with the “g” exchanged with the number “9”.

Table 3: Character substitution with a similar one

Orig	OCR	Count
g	9	40713
í	i	13992
D	B	8108
J	3	7617
i	l	5627
í	l	5270
£	t	5091
F	P	3042
I	l	2793
D	o	2636
o	a	2482
B	D	2017
L	u	1483
ri	n	1380
û	ú	1364
v	y	1302
m	rn	1292

Problems concerning the letter I

If we gather all substitutions concerning the letter “i” into one group, we can tell that among character changes this is the most common error. When looking at (Table 4) it can be easily understood that these characters are misrecognized because even for humans they may look really similar.

Substitution of numbers and letters

If a letter is substituted with a number, the original word can be, in most cases, easily reconstructed. It was interesting to see that in many cases the text processor was not able to do this. The word “hogy” was read as “ho9y” 7190 times. Which is a large number considering that the word “hogy” is included in the internal dictionary of the processing software.

5 Summary

In this paper we described a testbed which was used to test the accuracy of OCR software on Hungarian language documents. The results showed that for text

Table 4: "I"-related issues

Orig	OCR	Count
í	i	13992
I	i	10130
i	l	5627
Í	í	5574
í	l	5270
j	J	3283
I	l	2793
i	I	2637
l	1	1731
l	1	1639
1	l	1257
Í	I	1206

retrieval the most of the errors can be ignored, but there are some typical errors which have to be considered when working with such texts, such as the ones with accented characters or with the characters or marks with similar shape to letter I.

6 Future Plans

We still need to examine our results. We have a lot of search related lessons learned, and they provide a good base for search related products for digital libraries and data repositories.

Acknowledgments

The authors would like to express their thanks to László Kovács for his support as a scientific advisor. This paper was created in the scope and financial support of META-CONTENTUM [8] K+F project.

References

[1] J. D. Schanda, "Chapter 10 colorimetry," in *Handbook of Applied Photometry*, pp. 327–406, Springer Verlag, 1997.

[2] T. K. Ho, "A theory of multiple classifier systems and its application to visual word recognition," Tech. Rep. 92-12, 1992.

[3] C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[4] Wikipedia, "Moiré pattern." http://en.wikipedia.org/wiki/Moire.

[5] Due, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition-a survey," *Pattern Recognition*, vol. 29, pp. 641–662, April 1996.

[6] L.-M. Liu, Y. M. Babad, W. Sun, and K.-K. Chan, "Adaptive post-processing of ocr text via knowledge acquisition," in *CSC '91: Proceedings of the 19th annual conference on Computer Science*, (New York, NY, USA), pp. 558–569, ACM Press, 1991.

[7] G. Prószéky and B. Kis in *Számítógéppel - emberi nyelven*, SZAK, 1999.

[8] FreeSoft, "A meta-contentum k+f projekt." http://www.contentum.hu/hu/news/meta-contentum-kf.